

Learning Boolean Halfspaces with Small Weights from Membership Queries

Hasan Abasi

Department of Computer Science
Technion, Haifa, 32000

Ali Z. Abdi

Convent of Nazareth School
Grade 11, Abas 7, Haifa

Nader H. Bshouty

Department of Computer Science
Technion, Haifa, 32000

May 8, 2014

Abstract

We consider the problem of proper learning a Boolean Halfspace with integer weights $\{0, 1, \dots, t\}$ from membership queries only. The best known algorithm for this problem is an adaptive algorithm that asks $n^{O(t^5)}$ membership queries where the best lower bound for the number of membership queries is n^t [4].

In this paper we close this gap and give an adaptive proper learning algorithm with two rounds that asks $n^{O(t)}$ membership queries. We also give a non-adaptive proper learning algorithm that asks $n^{O(t^3)}$ membership queries.

1 Introduction

We study the problem of learnability of boolean halfspace functions from membership queries [2, 1]. Boolean halfspace is a function $f = [w_1x_1 + \dots + w_nx_n \geq u]$ from $\{0, 1\}^n$ to $\{0, 1\}$ where the *weights* w_1, \dots, w_n and the *threshold* u are integers. The function is 1 if the arithmetic sum $w_1x_1 + \dots + w_nx_n$ is greater or equal to u and zero otherwise. In the *membership query* model [2, 1] the learning algorithm has access to a *membership oracle* \mathcal{O}_f , for some *target function* f , that receives an assignment $a \in \{0, 1\}^n$ and returns $f(a)$. A *proper learning algorithm* for a class of functions C is an algorithm that has access to \mathcal{O}_f where $f \in C$ asks membership queries and returns a function g in C that is equivalent to f .

The problem of learning classes from membership queries only were motivated from many problems in different areas such as computational biology that arises in whole-genome (DNA) shotgun sequencing [8, 5, 10], DNA library screening [13], multiplex PCR method of genome physical mapping [11], linkage discovery problems of artificial intelligence [10], chemical reaction problem [3, 6, 7] and signature coding problem for the multiple access adder channels [9].

Another scenario that motivate the problem of learning Halfspaces is the following. Given a set of n similar looking objects of unknown weights (or any other measure), but from some class of weights W . Suppose we have a scale (or a measure instrument) that can only indicate whether the weight of any set of objects exceeds some unknown fixed threshold (or capacity). How many weighing do one needs in order to find the weights (or all possible weights) of the objects.

In this paper we study the problem of proper learnability of boolean halfspace functions with $t + 1$ different non-negative weights $W = \{0, 1, \dots, t\}$ from membership queries. The best known algorithm for this problem is an adaptive algorithm that asks $n^{O(t^5)}$ membership queries where the best lower bound for the number of membership queries is n^t [4].

In this paper we close the above gap and give an adaptive proper learning algorithm with two rounds that asks $n^{O(t)}$ membership queries. We also give a non-adaptive proper learning algorithm that asks $n^{O(t^3)}$ membership queries. All the algorithms in this paper runs in time that is linear in the membership query complexity.

Extending such result to non-positive weights is impossible. In [4] Aboud et. al. showed that in order to learn boolean Halfspace functions with weights $W = \{-1, 0, 1\}$, we need at least $O(2^{n-o(n)})$ membership queries. Therefore the algorithm that asks all the 2^n queries in $\{0, 1\}^n$ is optimal for this case. Shevchenko and Zolotykh [14] studied halfspace function over the domain $\{0, 1, \dots, k - 1\}^n$ and no constraints on the coefficients. They gave the lower bound $\Omega(\log^{n-2} k)$ lower bound for learning this class from membership queries. Hegedüs [12] prove the upper bound $O(\log^n k / \log \log n)$. For fixed n Shevchenko and Zolotykh [15] gave a polynomial time algorithm (in $\log k$) for this class.

This paper is organized as follows. In Section 2 we give some definitions and preliminary results. In Section 3 we show that any boolean halfspace with polynomially bounded coefficients can be expressed by an Automaton of polynomial size. A result that will be used in Section 4. In Section 4 we give the two round learning algorithm and the non-adaptive algorithm.

2 Definitions and Preliminary Results

In this section we give some definitions and preliminary results that will be used throughout the paper

2.1 Main Lemma

In this subsection we prove two main results that will be frequently used in this paper

For integers $t < r$ we denote $[t] := \{1, 2, \dots, t\}$, $[t]_0 = \{0, 1, \dots, t\}$ and $[t, r] = \{t, t+1, \dots, r\}$.

We first prove the following

Lemma 1. *Let $w_1, \dots, w_m \in [-t, t]$ where at least one $w_j \notin \{-t, 0, t\}$ and*

$$\sum_{i=1}^m w_i = r \in [-t+1, t-1].$$

There is a permutation $\phi : [m] \rightarrow [m]$ such that for every $j \in [m]$, $W_j := \sum_{i=1}^j w_{\phi(i)} \in [-t+1, t-1]$.

Proof. Since there is j such that $w_j \in [-t+1, t-1] \setminus \{0\}$ we can take $\phi(1) = j$. Then $W_1 = w_j \in [-t+1, t-1]$. If there is j_1, j_2 such that $w_{j_1} = t$ and $w_{j_2} = -t$ we set $\phi(2) = j_1$, $\phi(3) = j_2$ if $W_1 < 0$ and $\phi(2) = j_2$, $\phi(3) = j_1$ if $W_1 > 0$. We repeat the latter until there are either no more t or no more $-t$ in the rest of the elements.

Assume that we have chosen $\phi(1), \dots, \phi(k-1)$ such that $W_j \in [-t+1, t-1]$ for $j \in [k-1]$. We now show how to determine $\phi(k)$ so that $W_k \in [-t+1, t-1]$. If $W_{k-1} = \sum_{i=1}^{k-1} w_{\phi(i)} > 0$ and there is $q \notin \{\phi(1), \dots, \phi(k-1)\}$ such that $w_q < 0$ then we take $\phi(k) := q$. Then $W_k = W_{k-1} + w_q \in [-t+1, t-1]$. If $W_{k-1} < 0$ and there is $q \notin \{\phi(1), \dots, \phi(k-1)\}$ such that $w_q > 0$ then we take $\phi(k) := q$. Then $W_k = W_{k-1} + w_q \in [-t+1, t-1]$. If for every $q \notin \{\phi(1), \dots, \phi(k-1)\}$, $w_q > 0$ (resp. $w_q < 0$) then we can take an arbitrary order of the other elements and we get $W_{k-1} < W_k < W_{k+1} < \dots < W_m = r$ (resp. $W_{k-1} > W_k > W_{k+1} > \dots > W_m = r$). If $W_{k-1} = 0$ then there must be $q \notin \{\phi(1), \dots, \phi(k-1)\}$ such that $w_q \in [-t+1, t-1]$. This is because not both t and $-t$ exist in the elements that are not assigned yet. We then take $\phi(k) := q$.

This completes the proof. \square

We now prove the first main lemma

Lemma 2. Let $w_1, \dots, w_m \in [-t, t]$ and

$$\sum_{i=1}^m w_i = r \in [-t+1, t-1].$$

There is a partition S_1, S_2, \dots, S_q of $[m]$ such that

1. For every $j \in [q-1]$, $\sum_{i \in S_j} w_i = 0$.
2. $\sum_{i \in S_q} w_i = r$.
3. For every $j \in [q]$, $|S_j| \leq 2t-1$.
4. If $r \neq 0$ then $|S_q| \leq 2t-2$.

Proof. If $w_1, \dots, w_m \in \{-t, 0, t\}$ then r must be zero, and the number of non-zero elements is even and half of them are equal to t and the other half are equal to $-t$. Then we can take $S_i = \{-t, t\}$ or $S_i = \{0\}$ for all i . Therefore we may assume that at least one $w_j \notin \{-t, 0, t\}$.

By Lemma 1 we may assume w.l.o.g (by reordering the elements) that such that $W_j := \sum_{i=1}^j w_i \in [-t+1, t-1]$ for all $j \in [m]$. Let $W_0 = 0$. Consider $W_0, W_1, W_2, \dots, W_{2t-1}$. By the pigeonhole principle there is $0 \leq j_1 < j_2 \leq 2t-1$ such that $W_{j_2} = W_{j_1}$ and then $W_{j_2} - W_{j_1} = \sum_{i=j_1+1}^{j_2} w_i = 0$. We then take $S_1 = \{j_1+1, \dots, j_2\}$. Notice that $|S_1| = j_2 - j_1 \leq 2t-1$.

Since $\sum_{i \notin S_1} w_i = r$ we can repeat the above to find S_2, S_3, \dots . This can be repeated as long as $|[m] \setminus (S_1 \cup S_2 \cup \dots \cup S_h)| \geq 2t-1$. This proves 1-3.

We now prove 4. If $g := |[m] \setminus (S_1 \cup S_2 \cup \dots \cup S_h)| < 2t-1$ then define $S_{h+1} = [m] \setminus (S_1 \cup S_2 \cup \dots \cup S_h)$ and we get 4 for $q = h+1$. If $g = 2t-1$ then $W_0 = 0, W_1, W_2, \dots, W_{2t-1} = r$ and since $r \neq 0$ we must have $0 \leq j_1 < j_2 \leq 2t-1$ and $j_2 - j_1 < 2t-1$ such that $W_{j_2} = W_{j_1}$. Then define $S_{h+1} = \{j_1+1, \dots, j_2\}$, $S_{h+2} = [m] \setminus (S_1 \cup S_2 \cup \dots \cup S_{h+1})$ and $q = h+2$. Then $|S_{h+2}| \leq 2t-2$, $\sum_{i \in S_{h+1}} w_i = W_{j_2} - W_{j_1} = 0$ and $\sum_{i \in S_{h+2}} w_i = r$. \square

The following example shows that the bound $2t-2$ for the size of set in Lemma 2 is tight. Consider the $2t-2$ elements $w_1 = w_2 = \dots = w_{t-1} = t$ and $w_t = w_{t+1} = \dots = w_{2t-2} = -(t-1)$. The sum of any subset of elements is distinct. By adding the element $w_{2t-1} = -(t-1)$ it is easy to show that the bound $2t-1$ in the lemma is also tight.

Lemma 3. Let $(w_1, v_1), \dots, (w_m, v_m) \in [-t, t]^2$ and

$$\sum_{i=1}^m (w_i, v_i) = (r, s) \in [-t+1, t-1]^2.$$

There is $M \subseteq [m]$ such that

1. $\sum_{i \in M} (w_i, v_i) = (r, s)$.
2. $|M| \leq 8t^3 - 4t^2 - 2t + 1$.

Proof. Since $w_1, \dots, w_m \in [-t, t]$ and $\sum_{i=1}^m w_i = r \in [-t+1, t-1]$, by Lemma 2, there is a partition S_1, \dots, S_q of $[m]$ that satisfies the conditions 1–4 given in the lemma. Let $V_j = \sum_{i \in S_j} v_i$ for $j = 1, \dots, q$. We have

$$V_j \in [-t|S_j|, t|S_j|] \subseteq [-t(2t-1), t(2t-1)] \subset [-2t^2, 2t^2]$$

for $j = 1, \dots, q$ and

$$\sum_{i=1}^{q-1} V_i = s - V_q \in [-2t^2 + 1, 2t^2 - 1].$$

If $s - V_q = 0$ then for $M = S_q$ we have $|M| = |S_q| \leq 2t - 1 \leq 8t^3 - 4t^2 - 2t + 1$ and

$$\sum_{i \in M} (w_i, v_i) = \sum_{i \in S_q} (w_i, v_i) = (r, V_q) = (r, s).$$

Therefore we may assume that $s - V_q \neq 0$.

Consider V_1, V_2, \dots, V_{q-1} . By 4 in Lemma 2 there is a set $Q \subseteq [q-1]$ of size at most $2(2t^2) - 2 = 4t^2 - 2$ such that $\sum_{i \in Q} V_i = s - V_q$. Then for

$$M = S_q \cup \bigcup_{i \in Q} S_i$$

we have

$$|M| \leq (2t - 1) + (4t^2 - 2)(2t - 1) = 8t^3 - 4t^2 - 2t + 1$$

and

$$\begin{aligned} \sum_{i \in M} (w_i, v_i) &= \sum_{i \in S_q} (w_i, v_i) + \sum_{j \in Q} \sum_{i \in S_j} (w_i, v_i) \\ &= (r, V_q) + \sum_{j \in Q} (0, V_j) \\ &= (r, V_q) + (0, s - V_q) = (r, s). \end{aligned}$$

□

2.2 Boolean Functions

For a boolean function $f(x_1, \dots, x_n) : \{0, 1\}^n \rightarrow \{0, 1\}$, $1 \leq i_1 < i_2 < \dots < i_k \leq n$ and $\sigma_1, \dots, \sigma_k \in \{0, 1\}$ we denote by

$$f|_{x_{i_1}=\sigma_1, x_{i_2}=\sigma_2, \dots, x_{i_k}=\sigma_k}$$

the function f when fixing the variables x_{i_j} to σ_j for all $j \in [k]$. For $a \in \{0, 1\}^n$ we denote by $a|_{x_{i_1}=\sigma_1, x_{i_2}=\sigma_2, \dots, x_{i_k}=\sigma_k}$ the assignment a where each a_{i_j} is replaced by σ_j for all $j \in [k]$. We note here (and throughout the paper) that $f|_{x_{i_1}=\sigma_1, x_{i_2}=\sigma_2, \dots, x_{i_k}=\sigma_k}$ is a function from $\{0, 1\}^n \rightarrow \{0, 1\}$ with same variables x_1, \dots, x_n of f . Obviously

$$f|_{x_{i_1}=\sigma_1, x_{i_2}=\sigma_2, \dots, x_{i_k}=\sigma_k}(a) = f(a|_{x_{i_1}=\sigma_1, x_{i_2}=\sigma_2, \dots, x_{i_k}=\sigma_k}).$$

When $\sigma_1 = \dots = \sigma_k = \xi$ and $S = \{x_{i_1}, \dots, x_{i_k}\}$ we denote

$$f|_{S \leftarrow \xi} = f|_{x_{i_1}=\xi, x_{i_2}=\xi, \dots, x_{i_k}=\xi}.$$

In the same way we define $a|_{S \leftarrow \xi}$. We denote by $0^n = (0, 0, \dots, 0) \in \{0, 1\}^n$ and $1^n = (1, 1, \dots, 1) \in \{0, 1\}^n$. For two assignments $a \in \{0, 1\}^k$ and $b \in \{0, 1\}^j$ we denote by $ab \in \{0, 1\}^{k+j}$ the concatenation of the two assignments.

For two assignments $a, b \in \{0, 1\}^n$ we write $a \leq b$ if for every i , $a_i \leq b_i$. A boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is *monotone* if for every two assignments $a, b \in \{0, 1\}^n$, if $a \leq b$ then $f(a) \leq f(b)$. Recall that every monotone boolean function f has a unique representation as a reduced monotone DNF. That is, $f = M_1 \vee M_2 \vee \dots \vee M_s$ where each *monomial* M_i is an ANDs of input variables and for every monomial M_i there is a unique assignment $a^{(i)} \in \{0, 1\}^n$ such that $f(a^{(i)}) = 1$ and for every $j \in [n]$ where $a_j^{(i)} = 1$ we have $f(a^{(i)}|_{x_j=0}) = 0$. We call such assignment a *minterm* of the function f . Notice that every monotone DNF can be uniquely determined by its minterms.

We say that x_i is *relevant* in f if $f|_{x_i=0} \not\equiv f|_{x_i=1}$. Obviously, if f is monotone then x_i is relevant in f if there is an assignment a such that $f(a|_{x_i=0}) = 0$ and $f(a|_{x_i=1}) = 1$. We say that a is a *semiminterm* of f if for every $a_i = 1$ either $f(a|_{x_i=0}) = 0$ or x_i is not relevant in f .

For two assignments $a, b \in \{0, 1\}^n$ we define the *distance* between a and b as $wt(a+b)$ where wt is the Hamming weight and $+$ is the bitwise exclusive or of assignments. The set $B(a; d)$ is the set of all assignments that are of distance at most d from $a \in \{0, 1\}^n$.

2.3 Symmetric and Nonsymmetric

We say that a boolean function f is *symmetric* in x_i and x_j if for any $\xi_1, \xi_2 \in \{0, 1\}$ we have $f|_{x_i=\xi_1, x_j=\xi_2} \equiv f|_{x_i=\xi_2, x_j=\xi_1}$. Obviously, this is equivalent to $f|_{x_i=0, x_j=1} \equiv f|_{x_i=1, x_j=0}$. We say that f is *nonsymmetric* in x_i and x_j if it is not symmetric in x_i and x_j . This is equivalent to $f|_{x_i=0, x_j=1} \not\equiv f|_{x_i=1, x_j=0}$. We now prove

Lemma 4. *Let f be a monotone function. Then f is nonsymmetric in x_i and x_j if and only if there is a minterm a of f such that $a_i + a_j = 1$ (one is 0 and the other is 1) where $f(a|_{x_i=0, x_j=1}) \neq f(a|_{x_i=1, x_j=0})$.*

Proof. Since f is nonsymmetric in x_i and x_j we have $f|_{x_i=0, x_j=1} \not\equiv f|_{x_i=1, x_j=0}$ and therefore there is an assignment a' such that $f|_{x_i=0, x_j=1}(a') \neq f|_{x_i=1, x_j=0}(a')$. Suppose w.l.o.g. $f|_{x_i=0, x_j=1}(a') = 0$ and $f|_{x_i=1, x_j=0}(a') = 1$. Take a minterm $a \leq a'$ of $f|_{x_i=1, x_j=0}$. Notice that $a_i = a_j = 0$. Otherwise we can flip them to 0 without changing the value of the function $f|_{x_i=1, x_j=0}$ and then a is not a minterm. Then $f|_{x_i=1, x_j=0}(a) = 1$ and since $a \leq a'$, $f|_{x_i=0, x_j=1}(a) = 0$.

We now prove that $b = a|_{x_i=1, x_j=0}$ is a minterm of f . Since $b|_{x_i=0} = a|_{x_i=0, x_j=0} < a|_{x_i=0, x_j=1}$ we have $f(b|_{x_i=0}) < f(a|_{x_i=0, x_j=1}) = f|_{x_i=0, x_j=1}(a) = 0$ and therefore $f(b|_{x_i=0}) = 0$. For any $b_k = 1$ where $k \neq i$, since a is a minterm for $f|_{x_i=1, x_j=0}$, we have $f(b|_{x_k=0}) = f|_{x_i=1, x_j=0}(a|_{x_k=0}) = 0$. Therefore b is a minterm of f . \square

We write $x_i \sim_f x_j$ when f is symmetric in x_i and x_j and call \sim_f the symmetric relation of f . The following folklore result is proved for completeness

Lemma 5. *The relation \sim_f is an equivalence relation.*

Proof. Obviously, $x_i \sim_f x_i$ and if $x_i \sim_f x_j$ then $x_j \sim_f x_i$. Now if $x_i \sim_f x_j$ and $x_j \sim_f x_k$ then $f|_{x_i=\xi_1, x_j=\xi_2, x_k=\xi_3} \equiv f|_{x_i=\xi_2, x_j=\xi_1, x_k=\xi_3} \equiv f|_{x_i=\xi_2, x_j=\xi_3, x_k=\xi_1} \equiv f|_{x_i=\xi_3, x_j=\xi_2, x_k=\xi_1}$ and therefore $x_i \sim_f x_k$. \square

2.4 Properties of Boolean Halfspaces

A *Boolean Halfspace* function is a boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, $f = [w_1x_1 + w_2x_2 + \dots + w_nx_n \geq u]$ where w_1, \dots, w_n, u are integers, defined as $f(x_1, \dots, x_n) = 1$ if $w_1x_1 + w_2x_2 + \dots + w_nx_n \geq u$ and 0 otherwise. The numbers w_i , $i \in [n]$ are called the *weights* and u is called the *threshold*. The class HS is the class of all Boolean Halfspace functions. The class HS_t is the class of all Boolean Halfspace functions with weights $w_i \in [t]_0$ and the

class $\text{HS}_{[-t,t]}$ is the class of all Boolean Halfspace functions with weights $w_i \in [-t, t]$. The representation of the above Boolean Halfspaces are not unique. For example, $[3x_1 + 2x_2 \geq 2]$ is equivalent to $[x_1 + x_2 \geq 1]$. We will assume that

$$\text{There is an assignment } a \in \{0, 1\}^n \text{ such that } w_1 a_1 + \dots + w_n a_n = b \quad (1)$$

Otherwise we can replace b by the minimum integer $w_1 a_1 + \dots + w_n a_n$ where $f(a) = 1$ and get an equivalent function. Such a is called a *strong assignment* of f . If in addition a is a minterm then it is called a *strong minterm*.

The following lemma follows from the above definitions

Lemma 6. *Let $f \in \text{HS}_t$. We have*

1. *If a is strong assignment of f then a is semiminterm of f .*
2. *If all the variables in f are relevant then any semiminterm of f is a minterm of f .*

We now prove

Lemma 7. *Let $f = [w_1 x_1 + w_2 x_2 + \dots + w_n x_n \geq u] \in \text{HS}_t$. Then*

1. *If $w_1 = w_2$ then f is symmetric in x_1 and x_2 .*
2. *If f is symmetric in x_1 and x_2 then there are w'_1 and w'_2 such that $|w'_1 - w'_2| \leq 1$ and $f \equiv [w'_1 x_1 + w'_2 x_2 + w_3 x_3 + \dots + w_n x_n \geq u] \in \text{HS}_t$.*

Proof. If $w_1 = w_2$ then for any assignment $z = (z_1, z_2, \dots, z_n)$ we have $w_1 z_1 + w_2 z_2 + \dots + w_n z_n = w_1 z_2 + w_2 z_1 + \dots + w_n z_n$. Therefore, $f(0, 1, x_3, \dots, x_n) \equiv f(1, 0, x_3, \dots, x_n)$.

Suppose $w_1 > w_2$. It is enough to show that $f \equiv g := [(w_1 - 1)x_1 + (w_2 + 1)x_2 + w_3 x_3 + \dots + w_n x_n \geq u]$. Obviously, $f(x) = g(x)$ when $x_1 = x_2 = 1$ or $x_1 = x_2 = 0$. If $f(0, 1, x_3, \dots, x_n) \equiv f(1, 0, x_3, \dots, x_n)$ then $w_1 + w_3 x_3 + w_4 x_4 + \dots + w_n x_n \geq u$ if and only if $w_2 + w_3 x_3 + w_4 x_4 + \dots + w_n x_n \geq u$ and therefore $w_1 + w_3 x_3 + w_4 x_4 + \dots + w_n x_n \geq u$ if and only if $(w_1 - 1) + w_3 x_3 + w_4 x_4 + \dots + w_n x_n \geq u$ if and only if $(w_2 + 1) + w_3 x_3 + w_4 x_4 + \dots + w_n x_n \geq u$. \square

We now prove

Lemma 8. *Let $f \in \text{HS}_t$. Let a be any assignment such that $f(a) = 1$ and $f(a|_{x_i=0}) = 0$ for some $i \in [n]$. There is a strong assignment of f in $B(a, 2t - 2)$.*

Proof. Let $f = [w_1x_1 + \dots + w_nx_n \geq u]$. Since $f(a) = 1$ and $f|_{x_i=0}(a) = 0$, $a_i = 1$ and we have $w_1a_1 + w_2a_2 + \dots + w_na_n = u + u'$ where $t-1 \geq u' \geq 0$. If $u' = 0$ then $a \in B(a, 2t-2)$ is a strong assignment. So we may assume that $u' \neq 0$.

By (1) there is an assignment b where $w_1b_1 + w_2b_2 + \dots + w_nb_n = u$. Therefore $w_1(b_1 - a_1) + w_2(b_2 - a_2) + \dots + w_n(b_n - a_n) = -u'$. Since $w_i(b_i - a_i) \in [-t, t]$, by Lemma 2 there is $S \subseteq [n]$ of size at most $2t-2$ such that $\sum_{i \in S} w_i(b_i - a_i) = -u'$. Therefore

$$u = -u' + (u + u') = \sum_{i \in S} w_i(b_i - a_i) + \sum_{i=1}^n w_i a_i = \sum_{i \in S} w_i b_i + \sum_{i \notin S} w_i a_i.$$

Thus the assignment c where $c_i = b_i$ for $i \in S$ and $c_i = a_i$ for $i \notin S$ is a strong assignment of f and $c \in B(a, 2t-2)$. \square

The following will be used to find the relevant variables

Lemma 9. *Let $f \in \text{HS}_t$. Suppose x_k is relevant in f . Let a be any assignment such that $a_k = 1$, $f(a) = 1$ and $f(a|_{x_j=0}) = 0$ for some $j, k \in [n]$. There is $c \in B(a, 2t-2)$ such that $c_k = 1$, $f(c) = 1$ and $f(c|_{x_k=0}) = 0$.*

Proof. Let $f = [w_1x_1 + \dots + w_nx_n \geq u]$. Since $f(a) = 1$ and $f(a|_{x_j=0}) = 0$ we have $a_j = 1$ and $w_1a_1 + w_2a_2 + \dots + w_na_n = u + u'$ where $t-1 \geq u' \geq 0$. Let b a minterm of f such that $b_k = 1$. Since b is a minterm we have $w_1b_1 + w_2b_2 + \dots + w_nb_n = u + u''$ where $t-1 \geq u'' \geq 0$ and since $f(b|_{x_k=0}) = 0$ we also have $u'' - w_k < 0$. If $u'' = u'$ then we may take $c = a$. Therefore we may assume that $u'' \neq u'$.

Hence $\sum_{i=1, i \neq k}^n w_i(b_i - a_i) = u'' - u' \in [-t+1, t-1] \setminus \{0\}$. By Lemma 2 there is $S \subseteq [n] \setminus \{k\}$ of size at most $2t-2$ such that $\sum_{i \in S} w_i(b_i - a_i) = u'' - u'$. Therefore

$$u + u'' = \sum_{i \in S} w_i(b_i - a_i) + \sum_{i=1}^n w_i a_i = \sum_{i \in S} w_i b_i + \sum_{i \notin S} w_i a_i.$$

Thus the assignment c where $c_i = b_i$ for $i \in S$ and $c_i = a_i$ for $i \notin S$ satisfies $c_k = a_k = 1$ and $c \in B(a, 2t-2)$. Since $\sum_{i=1, i \neq k}^n w_i c_i = u + u'' - b_k < u$ we have $f(c|_{x_k=0}) = 0$. \square

The following will be used to find the order of the weights

Lemma 10. *Let $f \in \text{HS}_t$ be antisymmetric in x_1 and x_2 . For any minterm a of f of weight at least 2 there is $b \in B(a, 2t+1)$ such that $b_1 + b_2 = 1$ and $f|_{x_1=0, x_2=1}(b) \neq f|_{x_1=1, x_2=0}(b)$.*

Proof. Let $f = [w_1x_1 + \dots + w_nx_n \geq u]$. Assume w.l.o.g $w_1 > w_2$. By Lemma 4 there is a minterm $c = (1, 0, c_3, \dots, c_n)$ such that $f(c) = 1$ and $f(0, 1, c_3, \dots, c_n) = 0$. Then $W_1 := w_1 + w_3c_3 + \dots + w_nc_n = u + v$ where $0 \leq v \leq t - 1$ and $W_2 := w_2 + w_3c_3 + \dots + w_nc_n = u - z$ where $1 \leq z \leq t - 1$. In fact $-z = v - w_1 + w_2$. Since a is a minterm we have $W_3 := w_1a_1 + \dots + w_na_n = u + h$ where $0 \leq h \leq t - 1$. It is now enough to find $b \in B(a, 2t - 2)$ such that either

1. $b_1 = 1, b_2 = 0$ and $w_1b_1 + \dots + w_nb_n = u + v$, or
2. $b_1 = 0, b_2 = 1$ and $w_1b_1 + \dots + w_nb_n = u - z$.

This is because if $b_1 = 1, b_2 = 0$ and $w_1b_1 + \dots + w_nb_n = u + v$ (the other case is similar) then $f(1, 0, b_2, \dots, b_n) = 1$ and since $w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot a_3 \dots + w_na_n = u + v - w_1 + w_2 = u - z$ we have $f(0, 1, b_2, \dots, b_n) = 0$.

We now have four cases

Case I. $a_1 = 1$ and $a_2 = 0$: Then $W_1 - W_3 = w_3(c_3 - a_3) + \dots + w_n(c_n - a_n) = v - h \in [-t + 1, t - 1] \setminus \{0\}$. By Lemma 2 there is $S \subseteq [3, n]$ of size at most $2t - 1$ such that $\sum_{i \in S} w_i(c_i - a_i) = v - h$. Therefore

$$u + v = v - h + W_3 = \sum_{i \in S} w_i(c_i - a_i) + \sum_{i=1}^n w_i a_i = \sum_{i \in S} w_i c_i + \sum_{i \notin S} w_i a_i.$$

Now define b to be $b_i = c_i$ for $i \in S$ and $b_i = a_i$ for $i \notin S$. Since $1, 2 \notin S$ $b_1 = a_1 = 1$ and $b_2 = a_2 = 0$. Since $b \in B(a, 2t - 1) \subset B(a, 2t + 1)$ and b satisfies 1. the result follows for this case.

Case II. $a_1 = 0$ and $a_2 = 1$: Since a is of weight at least 2, we may assume w.l.o.g that $a_3 = 1$. Since a is a minterm $f(a) = 1$ and $f(a|_{x_3=0}) = 0$ and therefore for $a' = a|_{x_3=0}$ we have $W_4 := w_1a'_1 + w_2a'_2 + \dots + w_na'_n = u - h'$ where $1 \leq h' \leq t - 1$. Then $W_2 - W_4 = \sum_{i=3}^n w_i(c_i - a'_i) = h' - z \in [-t + 1, t - 1]$. By Lemma 2 there is $S \subseteq [3, n]$ of size at most $2t - 1$ such that $\sum_{i \in S} w_i(c_i - a'_i) = h' - z$. Therefore

$$u - z = h' - z + W_4 = \sum_{i \in S} w_i(c_i - a'_i) + \sum_{i=1}^n w_i a'_i = \sum_{i \in S} w_i c_i + \sum_{i \notin S} w_i a'_i.$$

Now define b to be $b_i = c_i$ for $i \in S$ and $b_i = a'_i$ for $i \notin S$. Since $1, 2 \notin S$ $b_1 = a'_1 = 0$ and $b_2 = a'_2 = 1$. Since $b \in B(a', 2t - 1) \subset B(a, 2t + 1)$ and b satisfies 2. the result follows for this case.

Case III. $a_1 = 1$ and $a_2 = 1$: Since a is a minterm $f(a) = 1$ and $f(a|_{x_1=0}) = 0$ and therefore for $a' = a|_{x_1=0}$ we have $W_4 := w_1a'_1 + w_2a'_2 + \dots + w_na'_n = u - h'$ where $1 \leq h' \leq t - 1$. We now proceed exactly as in Case II.

Case IV. $a_1 = 0$ and $a_2 = 0$: Since a is of weight at least 2 we may assume w.l.o.g that $a_3 = 1$. Since a is a minterm $f(a) = 1$ and $f(a|_{x_3=0}) = 0$ and therefore for $a' = a|_{x_3=0}$ we have $W_4 := a'_1 w_1 + a'_2 w_2 + \dots + a'_n w_n = u - h'$ where $1 \leq h' \leq t - 1$. If $f(a'|_{x_2=1}) = 0$ then proceed as in Case II to get $b \in B(a, 2t + 1)$ that satisfies 2. If $f(a'|_{x_1=1}) = 1$ then proceed as in Case I. Now the case where $f(a'|_{x_2=1}) = 1$ and $f(a'|_{x_1=1}) = 0$ cannot happen since $w_1 > w_2$. \square

The following will be used for the non-adaptive algorithm

Lemma 11. *Let $f, g \in \text{HS}_t$ be such that $f \not\equiv g$. For any minterm b of f there is $c \in B(b, 8t^3 + O(t^2))$ such that $f(c) + g(c) = 1$.*

Proof. Let $f = [w_1 x_1 + \dots + w_n x_n \geq u]$ and $g = [w'_1 x_1 + \dots + w'_n x_n \geq u']$. Since $f \not\equiv g$, there is $a' \in \{0, 1\}^n$ such that $f(a') = 1$ and $g(a') = 0$. Let $a \leq a'$ be a minterm of f . Then $f(a) = 1$ and since $a \leq a'$ we also have $g(a) = 0$. Therefore $w_1 a_1 + \dots + w_n a_n = u + r$ where $0 \leq r \leq t - 1$ and $w'_1 a_1 + \dots + w'_n a_n = u' - s$ for some integer $s \geq 1$. Since b is a minterm of f we have $w_1 b_1 + \dots + w_n b_n = u + r'$ where $0 \leq r' \leq t - 1$. If $g(b) = 0$ then take $c = b$. Otherwise, if for some $b_i = 1$, $g(b|_{x_i=0}) = 1$ then take $c = b|_{x_i=0}$. Therefore we may assume that b is also a minterm of g . Thus $w'_1 b_1 + \dots + w'_n b_n = u + s'$ where $0 \leq s' \leq t - 1$.

Consider the sequence Z_i , $i = 1, \dots, n + s - 1$ where $Z_i = (w_i(a_i - b_i), w'_i(a_i - b_i))$ for $i = 1, \dots, n$ and $Z_i = (0, 1)$ for $i = n + 1, \dots, n + s - 1$. Then

$$\sum_{i=1}^{n+s-1} Z_i = (r - r', -1 - s') \in [-t, t]^2.$$

By Lemma 3 there is a set $S \subseteq [n + s - 1]$ of size $8t^3 + O(t^2)$ such that $\sum_{i \in S} Z_i = (r - r', -1 - s')$. Therefore, there is a set $T \subseteq [n]$ of size at most $8t^3 + O(t^2)$ such that $\sum_{i \in T} Z_i = (r - r', -\ell - 1 - s')$ for some $\ell > 0$. Therefore

$$\sum_{i \in T} w_i(a_i - b_i) = r - r' \text{ and } \sum_{i \in T} w'_i(a_i - b_i) = -\ell - 1 - s'.$$

Define c such that $c_i = a_i$ for $i \in T$ and $c_i = b_i$ for $i \notin T$. Then

$$\sum_{i=1}^n w_i c_i = u + r \geq u \text{ and } \sum_{i=1}^n w'_i c_i = u' - \ell - 1 < u'.$$

Therefore $f(c) = 1$ and $g(c) = 0$. This gives the result. \square

3 Boolean Halfspace and Automata

In this section we show that functions in $\text{HS}_{[-t,t]}$ has an automaton representation of $\text{poly}(n, t)$ size.

Lemma 12. *Let $f_1, f_2, \dots, f_k \in \text{HS}_{[-t,t]}$ and $g : \{0, 1\}^k \rightarrow \{0, 1\}$. Then $g(f_1, \dots, f_k)$ can be represented with an Automaton of size $(2t)^k n^{k+1}$.*

Proof. Let $f_i = [w_{i,1}x_1 + \dots + w_{i,n}x_n \geq u_i]$, $i = 1, \dots, k$. Define the following automaton: The alphabet of the automaton is $\{0, 1\}$. The states are $S \subseteq [n]_0 \times [-tn, tn]^k$. The automaton has $n + 1$ levels. States in level i are connected only to states in level $i + 1$ for all $i \in [n]_0$. We denote by S_i the states in level i . We also have $S_i \subseteq \{i\} \times [-tn, tn]^k$ so the first entry of the state indicates the level that the state belongs to. The state $(0, (0, 0, \dots, 0))$ is the initial state and is the only state in level 0. That is $S_0 = \{(0, (0, 0, \dots, 0))\}$. We now show how to connect states in level i to states in level $i + 1$. Given a state $s = (i, (W_1, W_2, \dots, W_k))$ in S_i . Then the transition function for this state is

$$\delta((i, (W_1, W_2, \dots, W_k)), 0) = (i + 1, (W_1, W_2, \dots, W_k))$$

and

$$\delta((i, (W_1, W_2, \dots, W_k)), 1) = (i + 1, (W_1 + w_{1,i+1}, W_2 + w_{2,i+1}, \dots, W_k + w_{k,i+1})).$$

The accept states (where the output of the automaton is 1) are all the states $(n, (W_1, \dots, W_k))$ where $g([W_1 \geq u_1], [W_2 \geq u_2], \dots, [W_n \geq u_n]) = 1$. Here $[W_i \geq u_i] = 1$ if $W_i \geq u_i$ and zero otherwise. All other states are nonaccept states (output 0).

We now claim that the above automaton is equivalent to $g(f_1, \dots, f_k)$. The proof is by induction on n . The claim we want to prove is that the subautomaton that starts from state $s = (i, (W_1, W_2, \dots, W_k))$ computes a function g_s that is equivalent to the function $g(f_1^i, \dots, f_k^i)$ where $f_j^i = [w_{j,i+1}x_{i+1} + \dots + w_{j,n}x_n \geq u_j - W_j]$. This immediately follows from the fact that

$$g_s|_{x_{i+1}=0} \equiv g_{\delta(s,0)}, \quad \text{and} \quad g_s|_{x_{i+1}=1} \equiv g_{\delta(s,1)}.$$

It remains to prove the result for level n . The claim is true for the states at level n because

$$\begin{aligned} g(f_1^n, \dots, f_k^n) &= g([0 \geq u_1 - W_1], \dots, [0 \geq u_n - W_n]) \\ &= g([W_1 \geq u_1], [W_2 \geq u_2], \dots, [W_n \geq u_n]). \end{aligned}$$

This completes the proof. \square

Now the following will be used in the sequel

Lemma 13. *Let $f_1, f_2 \in \text{HS}_{[-t,t]}$. There is an algorithm that runs in time $t^2 n^3$ and decides whether $f_1 \equiv f_2$. If $f_1 \not\equiv f_2$ then the algorithm finds an assignment a such that $f_1(a) \neq f_2(a)$.*

Proof. We build an automaton for $f_1 + f_2$. If there is no accept state then $f_1 \equiv f_2$. If there is, then any path from the start state to an accept state defines an assignment a such that $f_1(a) \neq f_2(a)$. \square

4 Two Rounds and Non-adaptive Algorithm

In this section we give a two rounds algorithm for learning HS_t that uses $n^{O(t)}$ membership queries.

Let $f = [w_1 x_1 + \dots + w_n x_n \geq u]$. If there is a minterm of weight one then $0 \leq u \leq t$ and then all the minterms of f are of weight at most t . In this case we can find all the minterms in one round by asking all the assignments in $B(0, t)$ (all other assignments gives 0), finding all the relevant variables and the antisymmetric variables and move to the second round. Therefore we may assume that all the minterms of f are of weight at least two.

Consider the set

$$A_m = \bigcup_{i,j=0}^n B(0^i 1^{n-i-j} 0^j, m).$$

we now prove

Lemma 14. *Let $f \in \text{HS}_t$. The variable x_k is relevant in f if and only if there is $a \in A_{2t-2}$ such that $a_k = 1$, $a|_{x_k=0} \in A_{2t-1}$ and $f(a) \neq f(a|_{x_k=0})$.*

Proof. If x_k is relevant in f then $f \not\equiv 0, 1$ and therefore $f(0^n) = 0$ and $f(1^n) = 1$. Therefore there is an element a in the following sequence

$$0^n, 0^{k-1} 1 0^{n-k}, 0^{k-1} 1^2 0^{n-k-1}, \dots, 0^{k-1} 1^{n-k+1} 0, 0^{k-2} 1^{n-k+2}, \dots, 0 1^{n-1}, 1^n$$

and $j \in [n]$ such that $f(a) = 1$ and $f(a|_{x_j=0}) = 0$. Notice that $a_k = 1$ and therefore by Lemma 9 there is $c \in B(a, 2t-2)$ such that $c_k = 1$, $f(c) = 1$ and $f(c|_{x_k=0}) = 0$. Since $c|_{x_k=0} \in B(a, 2t-1)$, the result follows. \square

Therefore from the assignments in A_{2t-1} one can determine the relevant variables in f . This implies that we may assume w.l.o.g that all the variables are relevant. This can be done by just ignoring all the nonrelevant variables and projecting the relevant variables to new distinct variables y_1, \dots, y_m .

We now show

Lemma 15. *If all the variables in $f \in \text{HS}_t$ are relevant then there is a strong minterm $a \in A_{2t-2}$ of f .*

Proof. Follows from Lemma 8 and Lemma 6. \square

Lemma 16. *Let $f \in \text{HS}_t$ and suppose all the variables in f are relevant. Suppose f is antisymmetric in x_j and x_k . There is $b \in B(a, 4t-1)$ such that $b_1 + b_2 = 1$ and $f|_{x_j=0, x_k=1}(b) \neq f|_{x_j=1, x_k=0}(b)$.*

Proof. By Lemma 15 there is a minterm $a \in A_{2t-2}$ of f . Since $wt(a) > 1$, by Lemma 10 there is $b \in B(a, 2t-1)$ such that $b_1 + b_2 = 1$ and $f|_{x_j=0, x_k=1}(b) \neq f|_{x_j=1, x_k=0}(b)$. Since $b \in B(a, 2t+1) \subseteq A_{4t-1}$ the result follows. \square

Therefore from the assignments in A_{4t-3} one can find a permutation ϕ of the variables in f such that $f\phi = [w'_1x_1 + w'_2x_2 + \dots + w'_nx_n \geq u]$ and $w'_1 \leq w'_2 \leq \dots \leq w'_n$.

This completes the first round. We now may assume w.l.o.g that $f = [w_1x_1 + \dots + w_nx_n \geq u]$ and $1 \leq w_1 \leq w_2 \leq \dots \leq w_n \leq t$ and all the variables are relevant. The goal of the second round is to find $w_i \in [1, t]$ and $u \in [0, nt]$. Since we know that $1 \leq w_1 \leq w_2 \leq \dots \leq w_n \leq t$ we have

$$\binom{n+t-1}{t-1} nt \leq n^{t+1}$$

choices. That is at most n^{t+1} possible functions in HS_t . For every two such functions f_1, f_2 we use Lemma 13 to find out if $f_1 \equiv f_2$ and if not to find an assignment a such that $f_1(a) \neq f_2(a)$. This takes time

$$\binom{n^{t+1}}{2} t^2 n^3 \leq n^{2t+7}.$$

Let B the set of all such assignments. Then $|B| \leq n^{2t+2}$. In the second round we ask membership queries with all the assignments in B .

Now notice that if $f_1(a) \neq f_2(a)$ then either $f(a) \neq f_1(a)$ or $f(a) \neq f_2(a)$. This shows that the assignments in B eliminates all the functions that are not equivalent to the target and all the remaining functions are equivalent to the target.

Now using Lemma 11 one can replace the set B by $B(b, 8t^3 + O(t^2))$ for any minterm b of f . This change the algorithm to a non-adaptive algorithm.

References

- [1] M. Aigner. Combinatorial Search. Wiley Teubner Series on Applicable Theory in Computer Science. Teubner, Stuttgart. (1988).
- [2] D. Angluin. Queries and Concept Learning. Machine Learning 2(4), pp. 319–342, (1987).
- [3] A. Alon, V. Asodi. Learning a Hidden Subgraph. SIAM J. Discrete Math. 18(4): 697-712 (2005)
- [4] E. Abboud, N. Agha, N. H. Bshouty, N. Radwan, F. Saleh. Learning Threshold Functions with Small Weights Using Membership Queries. COLT 1999: 318-322
- [5] N. Alon, R. Beigel, S. Kasif, S. Rudich, B. Sudakov. Learning a Hidden Matching. SIAM J. Comput. 33(2): 487-501 (2004).
- [6] D. Angluin, J. Chen. Learning a Hidden Hypergraph. Journal of Machine Learning Research 7, pp. 2215-2236 (2006).
- [7] D. Angluin, J. Chen. Learning a hidden graph using $O(\log n)$ queries per edge. J. Comput. Syst. Sci. 74(4): 546-556 (2008).
- [8] R. Beigel, N. Alon, S. Kasif, M. Serkan Apaydin, L. Fortnow. An optimal procedure for gap closing in whole genome shotgun sequencing. RECOMB 2001, pp. 22-30
- [9] E. Biglieri, L. Györfi. Multiple Access Channels: Theory and Practice. IOS Press, (2007).
- [10] S-S. Choi, J. H. Kim. Optimal query complexity bounds for finding graphs. Artif. Intell. 174(9-10): 551-569 (2010).
- [11] V. Grebinski, G. Kucherov. Reconstructing a Hamiltonian Cycle by Querying the Graph: Application to DNA Physical Mapping. Discrete Applied Mathematics 88(1-3), pp. 147-165 (1998).
- [12] T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. Proceedings of the 8th Annual ACM Conference on Computational Learning Theory (COLT'95). ACM Press New York, pp.108–117. (1995).

- [13] H. Q. Ngo and D-Z. Du. A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening. DIMACS Series in Discrete Mathematics and Theoretical Computer Science.
- [14] V. N. Shevchenko, N. Yu. Zolotykh. Lower Bounds for the Complexity of Learning Half-Spaces with Membership Queries. ALT 1998, pp. 61-71.
- [15] N.Yu. Zolotykh and V. N. Shevchenko. Deciphering threshold functions of k-valued logic. Discrete Analysis and Operations Research. Novosibirsk 2 (3) (1995) pp. 18–23. English transl.: Korshunov, A. D. (ed.): Operations Research and Discrete Analysis. Kluwer Ac. Publ. Netherlands (1997) pp. 321–326.